AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application;

1. (Previously Presented) An encoder comprising:

information sequence generating means for generating a first information sequence by inserting a first DC control bit into an input information sequence at predetermined intervals and for generating a second information sequence by inserting a second DC control bit different from the first DC control bit into the input information sequence at the predetermined intervals;

first code converting means for generating a first provisional code sequence by making a code conversion of the first information sequence generated by the information sequence generating means at a conversion ratio of an information word length m to an code word length n;

second code converting means for generating a second provisional code sequence by making a code conversion of the second information sequence generated by the information sequence generating means at the conversion ratio of the information word length m to the code word length n; and

selecting means for selecting either the first provisional code sequence generated by the first code converting means or the second provisional code sequence generated by the second code converting means, wherein

the first and second code converting means use a coding rule by which, in case the coding rule is represented by a finite-state code conversion table, code words are assigned to information words so that a two's complement of a sum of coding bits included in the first provisional code sequence is always different from a two's complement of a sum of coding bits included in the second provisional code sequence when a first code state of the first provisional code sequence encoded starting with a predetermined original state is identical to a second code state of the second provisional code sequence encoded starting with the predetermined original state.

2. (Previously Presented) The encoder as set forth in claim 1, wherein the first and second code converting means makes a code conversion with an information word length m of 2 bits and in code conversion units of 2 bits.

3. (Currently Amended) The encoder as set forth in claim 2, wherein the first and second code converting means uses a coding rule under which the code words are assigned to the information words so that when the first code state is identical to the second code state, the two's complement of the sum of coding bits included in the first provisional code sequence is always different from the two's complement of the sum of coding bits included in the second provisional code sequence whether the first and second DC control bits are inserted at the first bit of the code conversion unit or the second bit of the code conversion unit by the information sequence generating means.

4. (Previously Presented) The encoder as set forth in claim 2, wherein the first and second code converting means uses a coding rule under which the code words are assigned to the information words so that when the first code state is identical to the second code state, the two's complement of the sum of coding bits included in the first provisional code sequence is always different from the two's complement of the sum of coding bits included in the second provisional code sequence when the first and second DC control bits are inserted at the first bit of the code conversion unit by the information sequence generating means.

5. (Previously Presented) The encoder as set forth in claim 2, wherein the first and second code converting means uses a coding rule by which the code words are assigned to the information words so that when the first code state is identical to the second code state, the two's complement of the sum of coding bits included in the first provisional code sequence is always different from the two's complement of the sum of coding bits included in the second provisional code sequence when the first and second DC control bits are inserted at the second bit of the code conversion unit by the information sequence generating means.

6. (Previously Presented) The encoder as set forth in claim 1, wherein the first and second code converting means is

designed to encode information with an information word length m of one block and in units of one block, and the first and second code converting means using a coding rule under which, in case the coding rule is represented by a look-ahead code conversion table, code words are assigned to information words so that the two's complement of a sum of coding bits included in an information sequence is different from the two's complement of a sum of coding bits included in a provisional code sequence when the number of blocks is odd while the two's complement of a sum of coding bits included in an information sequence coincides with the two's complement of a sum of coding bits included in a provisional code sequence when the number of blocks is even.

7. (Previously Presented) The encoder as set forth in claim 1, wherein the first and second code converting means is designed to make a code conversion with an information word length of 2 bits taken as one block and in units of one block according to a coding rule in which the information word has a length m of 2 bits, the code word has a length n of 3 bits and a maximum run is limited to 7.

8. (Original) The encoder as set forth in claim 7, wherein the first and second code converting means uses a coding rule in which, in case the coding rule is represented by a finite-state code conversion table, the number of states is seven.

9. (Previously Presented) The encoder as set forth in claim 7, wherein the first and second code converting means is designed to make a code conversion so that a decoded code word constraint length is 3 blocks.

10. (Currently Amended) The encoder as set forth in claim 1, wherein the first and second code converting means uses a coding rule in which a minimum run is limited to one while a maximum run is limited to seven and a maximum number of minimum runs in a sequence of code words is five and, in case the coding rule is represented by a code conversion table, a number of states is eight.

11. (Previously Presented) The encoder as set forth in claim 1, further comprising a code select signal generating means for calculating a digital-sum variation (DSV) of the first provisional code sequence and a DSV of the second provisional code sequence and generating a code select signal indicative of the provisional code sequence whose DSV is smaller, the selecting means selecting the provisional code sequence whose DSV is smaller according to the code select signal.

12. (Previously Presented) A coding method comprising the steps of:

generating a first information sequence by inserting a first DC control bit into an input information sequence at predetermined intervals and generating a second information sequence by inserting a second DC control bit different from the first DC control bit into the input information sequence at the predetermined intervals;

generating a first provisional code sequence by making a code conversion of the first information sequence generated in the information sequence generating step at a conversion ratio of an information word length m to an code word length n;

generating a second provisional code sequence by making a code conversion of the second information sequence generated in the information sequence generating step at the conversion ratio of the information word length m to the code word length n; and

selecting either the first provisional code sequence generated in the first code generating step or the second provisional code sequence generated in the second code generating step, wherein

the first and second code converting steps use a coding rule by which, in case the coding rule is represented by a finite-state code conversion table, code words are assigned to information words so that a two's complement of a sum of coding bits included in the first provisional code sequence is always different from a two's complement of a sum of coding bits included in the second provisional code sequence when a first code state of the first provisional code sequence encoded starting with a predetermined original state, is identical to a second code state of the second provisional code sequence encoded starting with the predetermined original state.

13. (Previously Presented) A recording medium having recorded therein a coding program to cause a computer to convert an m-bit information word continuously into an n-bit code word, the program comprising the steps of:

generating a first information sequence by inserting a first DC control bit into an input information sequence at predetermined intervals and generating a second information sequence by inserting a second DC control bit different from the first DC control bit into the input information sequence at the predetermined intervals;

generating a first provisional code sequence by making a code conversion of the first information sequence generated in the information sequence generating step at a conversion ratio of an information word length m to an code word length n;

generating a second provisional code sequence by making a code conversion of the second information sequence generated in the information sequence generating step at the conversion ratio of the information word length m to the code word length n; and

selecting either the first provisional code sequence generated in the first code generating step or the second provisional code sequence generated in the second code generating step, wherein

the first and second code converting steps uses a coding rule by which, in case the coding rule is represented by a finite-state code conversion table, code words are assigned to information words so that a two's complement of a sum of coding bits included in the first provisional code sequence is always different from of two's complement of a sum of coding bits included in the second provisional code sequence when a first code state of the first provisional code sequence encoded starting with a predetermined original state, is identical to a second code state of the second provisional code sequence encoded starting with the predetermined original state.

14. (Previously Presented) A coding program for converting an m-bit information word continuously into an n-bit code word, the program comprising the steps of:

generating a first information sequence by inserting a first DC control bit into an input information sequence at predetermined intervals and generating a second information sequence by inserting a second DC control bit different from the first DC control bit into the input information sequence at the predetermined intervals;

generating a first provisional code sequence by making a code conversion of the first information sequence generated in the information sequence generating step at a conversion ratio of an information word length m to an code word length n;

generating a second provisional code sequence by making a code conversion of the second information sequence generated in the information sequence generating step at the conversion ratio of the information word length m to the code word length n; and

selecting either the first provisional code sequence generated in the first code generating step or the second provisional code sequence generated in the second code generating step,

the first and second code converting steps uses a coding rule by which, in case the coding rule is represented by a finite-state code conversion table, code words are assigned to information words so that a two's complement of a sum of coding bits included in the first provisional code sequence is always different from a two's complement of a sum of coding bits included in the second provisional code sequence when a first code state of the first provisional code sequence encoded starting with a predetermined original state, is identical to a second code state of the second provisional code sequence encoded starting with the predetermined original state.

15. - 20. (Cancelled)

21. (New) The coding method as set forth in claim 12, wherein the making a code conversion of the first and second information sequences makes a code conversion with an information word length m of 2 bits and in code conversion units of 2 bits.

22. (New) The coding method as set forth in claim 21,

wherein steps of generating the first and second provisional code sequences use a coding rule under which the code words are assigned to the information words so that when the first code state is identical to the second code state, the two's complement of the sum of coding bits included in the first provisional code sequence is always different from the two's complement of the sum of coding bits included in the second provisional code sequence whether the first and second DC control bits are inserted at the first bit of the code conversion unit or the second bit of the code conversion unit by the step of generating a first information sequence.

23. (New) The coding method as set forth in claim 21, wherein the steps of generating the first and second provisional code sequences use a coding rule under which the code words are assigned to the information words so that when the first code state is identical to the second code state, the two's complement of the sum of coding bits included in the first provisional code sequence is always different from the two's complement of the sum of coding bits included in the second provisional code sequence when the first and second DC control bits are inserted at the first bit of the code conversion unit by the step of generating a first information sequence.

24. (New) The coding method as set forth in claim 21, wherein the step of generating the first and second provisional code sequences use a coding rule by which the code

words are assigned to the information words so that when the first code state is identical to the second code state, the two's complement of the sum of coding bits included in the first provisional code sequence is always different from the two's complement of the sum of coding bits included in the second provisional code sequence when the first and second DC control bits are inserted at the second bit of the code conversion unit by the step of generating a first information sequence.

25. (New) The coding method as set forth in claim 12, wherein the making a code conversion of the first and second information sequences is designed to encode information with an information word length m of one block and in units of one block, and the steps of generating the first and second provisional code sequences use a coding rule under which, in case the coding rule is represented by a look-ahead code conversion table, code words are assigned to information words so that the two's complement of a sum of coding bits included in an information sequence is different from the two's complement of a sum of coding bits included in a provisional code sequence when the number of blocks is odd while the two's complement of a sum of coding bits included in an information sequence coincides with the two's complement of a sum of coding bits included in a provisional code sequence when the number of blocks is even.

26. (New) The coding method as set forth in claim 12, wherein the making a code conversion of the first and second information sequences is designed to make a code conversion with an information word length of 2 bits taken as one block and in units of one block according to a coding rule in which the information word has a length m of 2 bits, the code word has a length n of 3 bits and a maximum run is limited to 7.

27. (New) The coding method as set forth in claim 26, wherein the making a code conversion of the first and second information sequences uses a coding rule in which, in case the coding rule is represented by a finite-state code conversion table, the number of states is seven.

28. (New) The coding method as set forth in claim 26, wherein the making a code conversion of the first and second information sequences is designed to make a code conversion so that a decoded code word constraint length is 3 blocks.

29. (New) The coding method as set forth in claim 12, wherein the making a code conversion of the first and second information sequences uses a coding rule in which a minimum run is limited to one while a maximum run is limited to seven and a maximum number of minimum runs in a sequence of code words is five and, in case the coding rule is represented by a code conversion table, a number of states is eight.

30. (New) The coding method as set forth in claim 12,

further comprising the steps of calculating a digital-sum variation (DSV) of the first provisional code sequence and a DSV of the second provisional code sequence and generating a code select signal indicative of the provisional code sequence whose DSV is smaller, and selecting the provisional code sequence whose DSV is smaller according to the code select signal.

31. (New) The recording medium as set forth in claim 13, wherein the making the code conversion of the first and second information sequences makes a code conversion with an information word length m of 2 bits and in code conversion units of 2 bits.

32. (New) The recording medium as set forth in claim 31, wherein the steps of generating the first and second provisional code sequences use a coding rule under which the code words are assigned to the information words so that when the first code state is identical to the second code state, the two's complement of the sum of coding bits included in the first provisional code sequence is always different from the two's complement of the sum of coding bits included in the second provisional code sequence whether the first and second DC control bits are inserted at the first bit of the code conversion unit or the second bit of the code conversion unit by the step of generating a first information sequence.

33. (New) The recording medium as set forth in claim 31, wherein the steps of generating the first and second provisional code sequences use a coding rule under which the code words are assigned to the information words so that when the first code state is identical to the second code state, the two's complement of the sum of coding bits included in the first provisional code sequence is always different from the two's complement of the sum of coding bits included in the second provisional code sequence when the first and second DC control bits are inserted at the first bit of the code conversion unit by the step of generating a first information sequence.

34. (New) The recording medium as set forth in claim 31, wherein the steps of generating the first and second provisional code sequences use a coding rule by which the code words are assigned to the information words so that when the first code state is identical to the second code state, the two's complement of the sum of coding bits included in the first provisional code sequence is always different from the two's complement of the sum of coding bits included in the second provisional code sequence when the first and second DC control bits are inserted at the second bit of the code conversion unit by the step of generating a first information sequence.

35. (New) The recording medium as set forth in claim 13, wherein the making a code conversion of the first and second

information sequences is designed to encode information with an information word length m of one block and in units of one block, and the steps of generating the first and second provisional code sequences use using a coding rule under which, in case the coding rule is represented by a look-ahead code conversion table, code words are assigned to information words so that the two's complement of a sum of coding bits included in an information sequence is different from the two's complement of a sum of coding bits included in a provisional code sequence when the number of blocks is odd while the two's complement of a sum of coding bits included in an information sequence coincides with the two's complement of a sum of coding bits included in a provisional code sequence when the number of blocks is even.

36. (New) The recording medium as set forth in claim 13, wherein the making a code conversion of the first and second information sequences is designed to make a code conversion with an information word length of 2 bits taken as one block and in units of one block according to a coding rule in which the information word has a length m of 2 bits, the code word has a length n of 3 bits and a maximum run is limited to 7.

37. (New) The recording medium as set forth in claim 36, wherein the making a code conversion of the first and second information sequences uses a coding rule in which, in case the coding rule is represented by a finite-state code conversion

table, the number of states is seven.

38. (New) The recording medium as set forth in claim 36, wherein the making a code conversion of the first and second information sequences is designed to make a code conversion so that a decoded code word constraint length is 3 blocks.

39. (New) The recording medium as set forth in claim 13, wherein the making a code conversion of the first and second information sequences uses a coding rule in which a minimum run is limited to one while a maximum run is limited to seven and a maximum number of minimum runs in a sequence of code words is five and, in case the coding rule is represented by a code conversion table, a number of states is eight.

40. (New) The recording medium as set forth in claim 13, further comprising the steps of calculating a digital-sum variation (DSV) of the first provisional code sequence and a DSV of the second provisional code sequence and generating a code select signal indicative of the provisional code sequence whose DSV is smaller, and selecting the provisional code sequence whose DSV is smaller according to the code select signal.

41. (New) The coding program as set forth in claim 14, wherein the making the code conversion of the first and second information sequences makes a code conversion with an information word length m of 2 bits and in code conversion

units of 2 bits.

42. (New) The coding program as set forth in claim 41, wherein the steps of generating the first and second provisional code sequences use a coding rule under which the code words are assigned to the information words so that when the first code state is identical to the second code state, the two's complement of the sum of coding bits included in the first provisional code sequence is always different from the two's complement of the sum of coding bits included in the second provisional code sequence whether the first and second DC control bits are inserted at the first bit of the code conversion unit or the second bit of the code conversion unit by the step of generating a first information sequence.

43. (New) The coding program as set forth in claim 41, wherein the steps of generating the first and second provisional code sequences use a coding rule under which the code words are assigned to the information words so that when the first code state is identical to the second code state, the two's complement of the sum of coding bits included in the first provisional code sequence is always different from the two's complement of the sum of coding bits included in the second provisional code sequence when the first and second DC control bits are inserted at the first bit of the code conversion unit by the step of generating a first information sequence.

44. (New) The coding program as set forth in claim 41, wherein the steps of generating the first and second provisional code sequences use a coding rule by which the code words are assigned to the information words so that when the first code state is identical to the second code state, the two's complement of the sum of coding bits included in the first provisional code sequence is always different from the two's complement of the sum of coding bits included in the second provisional code sequence when the first and second DC control bits are inserted at the second bit of the code conversion unit by the step of generating a first information sequence.

44. (New) The coding program as set forth in claim 14, wherein the making a code conversion of the first and second information sequences is designed to encode information with an information word length m of one block and in units of one block, and the steps of generating the first and second provisional code sequences use a coding rule under which, in case the coding rule is represented by a look-ahead code conversion table, code words are assigned to information words so that the two's complement of a sum of coding bits included in an information sequence is different from the two's complement of a sum of coding bits included in a provisional code sequence when the number of blocks is odd while the two's complement of a sum of coding bits included in an information sequence coincides with the two's complement of a sum of

coding bits included in a provisional code sequence when the number of blocks is even.

46. (New) The coding program as set forth in claim 14, wherein the making a code conversion of the first and second information sequences is designed to make a code conversion with an information word length of 2 bits taken as one block and in units of one block according to a coding rule in which the information word has a length m of 2 bits, the code word has a length n of 3 bits and a maximum run is limited to 7.

47. (New) The coding program as set forth in claim 46, wherein the making a code conversion of the first and second information sequences uses a coding rule in which, in case the coding rule is represented by a finite-state code conversion table, the number of states is seven.

48. (New) The coding program as set forth in claim 46, wherein the making a code conversion of the first and second information sequences is designed to make a code conversion so that a decoded code word constraint length is 3 blocks.

49. (New) The coding program as set forth in claim 14, wherein the making a code conversion of the first and second information sequences uses a coding rule in which a minimum run is limited to one while a maximum run is limited to seven and a maximum number of minimum runs in a sequence of code

words is five and, in case the coding rule is represented by a code conversion table, a number of states is eight.

50. (New) The coding program as set forth in claim 14, further comprising the steps of calculating a digital-sum variation (DSV) of the first provisional code sequence and a DSV of the second provisional code sequence and generating a code select signal indicative of the provisional code sequence whose DSV is smaller, and selecting the provisional code sequence whose DSV is smaller according to the code select signal.